

Abstract In this paper we model a situation where several wireless body sensor networks (WBSN) compete for occupation of a number of frequency channels. Each channel can host at most one WBSN with satisfactory performance and WBSNs have the ability to change their operating channel, subject to the constraint that they can only monitor the performance or occupancy of their current channel but not of any other channel. We consider a number of randomized schemes for changing the frequency channels and present and evaluate Markov chain models for these, building on a “balls-in-bins” approach.

Keywords Wireless Body Sensor Networks; Frequency Allocation; Balls and Bins; Markov Chains

Autonomous Allocation of Wireless Body-Sensor Networks to Frequency Channels: Modeling with Repeated “Balls-In-Bins” Experiments

Andreas Willig

the date of receipt and acceptance should be inserted later

1 Introduction

Wireless body sensor networks (WBSNs) have received lots of interest recently, because they enable a range of applications in health and well-being [21], [6], [2], [34]. A number of technologies are considered for WBSNs, including the IEEE 802.15.4 [20] and IEEE 802.15.6 [19] standards. On the physical layer the IEEE 802.15.4 standard specifies three different frequency bands that can be used. Two of these are sub-divided into a number of frequency channels, and a WBSN at any time operates on only one of these channels. For example, the popular 2.4 GHz ISM range is sub-divided into 16 channels of 5 MHz width each. Similarly, the IEEE 802.15.6 standard supports a number of different frequency ranges in its narrowband physical layer, and all these frequency ranges are sub-divided into several channels (79 in the 2.4 GHz range). In both standards it is foreseen that a WBSN does not routinely hop over the channels but rather picks a channel and stays there.

We consider situations where several independent WBSNs are forced to operate in close proximity to each other (for example when many people congregate in a sports stadium) and have to share wireless resources. In particular, each WBSN must decide on a frequency channel on which it operates. To detect co-location with other WBSNs in the same channel, we assume that WBSNs have the capability to monitor their own performance, e.g. packet loss rate, delay, number of retransmissions or other relevant indicators. In response to performance degradations from competing WBSNs, a WBSN can theoretically adjust a wide range of operational parameters (e.g. transmit power or data generation rate), but in this paper we focus on how WBSNs can make autonomous decisions about their frequency channel. In previous work we have demonstrated how such a capability can be practically implemented for IEEE 802.15.4 [40], [26].

We consider an abstract version of this setting and seek insights into the operation and performance of simple probabilistic strategies by which individual WBSNs

A. Willig is with the Department of Computer Science and Software Engineering, University of Canterbury, New Zealand. E-mail: andreas.willig@canterbury.ac.nz

decide autonomously when to switch to another channel, with the goal of quickly settling into a channel assignment satisfying the communications needs of the involved WBSNs. Suppose we are given N different channels and K different WBSNs. For simplicity we make the assumption that one channel can satisfy the communication needs of no more than one WBSN. When two or more WBSNs are present on a channel (which we refer to as a **collision**) then all WBSNs will experience insufficient transmission quality (please note that the precise definition of transmission quality does not matter for our purposes). We assume that all colliding WBSNs notice this immediately and might subsequently decide to jump to another channel. Under these assumptions there are two fundamentally different cases to consider:

- When $K \leq N$, i.e. the number of WBSNs does not exceed the number of channels, then it is clearly possible to find an allocation of WBSNs to channels that satisfies everyone – we will refer to such an allocation as a “noncolliding state”. A key performance metric is the average time to reach a non-colliding state.
- When $K > N$ then no non-colliding state exists and the choice of performance measure is not so clear-cut. Ideally, there should also be some degree of fairness among the WBSNs, i.e. each WBSN should experience satisfactory channel quality at least for some fraction of time and should be able to communicate successfully fairly frequently. The problem then becomes similar to load-balancing problems.

The main focus of this paper is on the case $K \leq N$, and we develop and analyse algorithms aiming to minimize the average time until there are no collisions. We will also assess the performance of one of these algorithms in the case $K > N$.

1.1 Contributions

We make four main contributions:

- We investigate a simple probabilistic algorithm, called the **restrained-jumping scheme** (RJS), in which a colliding WBSN jumps with a pre-specified probability p out of its current channel into one of the other $N - 1$ channels. This algorithm is somewhat similar to channel allocation and load balancing algorithms in the literature [31], [13]. We assume that $K \leq N$ and investigate the average time to reach a non-colliding state. We argue that this “blind hopping” approach is appropriate for situations where a WBSN colliding in its current channel has no information about the occupancy of other channels (which makes sense when the other WBSNs *also* jump randomly), so will have no preferred channel to jump to. We model this system as a “balls-in-bins” system [16] with repeated throws of a subset of the balls (the colliding ones), and develop a new and exact discrete-time Markov chain model [29], [22] for it. With this model, finding the average time to reach a non-colliding state (henceforth called **hitting time** in the paper) requires solving a linear equation system. We describe a method to reduce its dimension substantially. Furthermore, we provide a range of numerical results and identify important trends.

- The RJS algorithm does not require individual WBSNs to keep any state besides knowing the value of the jumping probability p . We propose a modified probabilistic algorithm called **RJS-OB** (where OB stands for “one-bit”), in which each WBSN uses a state of one additional bit to modify its jumping probability when colliding with other WBSNs, to the effect that a WBSN which has found a free channel and has “settled” there will now show some reluctance to leave by adopting a smaller jumping probability. We evaluate the performance of this algorithm through simulations for the case $K \leq N$ and show that it offers substantially reduced times to reach non-colliding states as compared to the best RJS algorithm. To the best of our knowledge, the RJS-OB algorithm is new.
- We consider a corner case of the RJS-OB scheme called the **sticky scheme**, in which WBSNs that have occupied a channel for at least one time slot as a sole occupant refuse to ever jump away from it. For $K \leq N$ this scheme shows the best performance in terms of the average time to reach a non-colliding state, but for $K > N$ it will be very unfair, allowing some WBSNs to grab a slot forever while the other ones have to jump around eternally. We develop a Markov model for this scheme (valid for $K \leq N$), establish some of its properties and provide numerical and simulation-based evidence that the average time to reach a non-colliding state increases only in an approximately linear fashion for large N and $K \leq N$. The sticky scheme thus substantially outperforms the RJS-OB and the best possible RJS scheme.
- While being designed to minimize the hitting time in the case $K \leq N$, the RJS-OB scheme is also applicable when $K > N$. We consider its fairness and the percentage of successful slots for a range of its parameters, and argue that it can be configured in a way that limits unfairness and approaches the performance of a pure balls-in-bins allocation (i.e. an allocation where each ball picks its bin independently and uniformly) from above, whereas by adding some unfairness the RJS-OB scheme achieves better performance.

We argue that the developed models and algorithms are of significant practical interest and are attractive candidates for the considered scenario. The algorithms have some key advantages:

- They are completely distributed and require no communication between WBSNs.
- They are guaranteed to converge to a non-colliding state with probability one, assuming it exists (i.e. $K \leq N$ holds), and we hypothesize that the sticky scheme asymptotically achieves this on average in a time linear in N .
- They do not require an individual WBSN to observe any other channel than its own, which is useful when data communication within a WBSN can happen at any time and no node can take a leave of absence to monitor other channels.

Clearly, we expect for our algorithms that their hitting time will generally be inferior to the times reachable with algorithms involving explicit negotiations and signaling between WBSNs, for example a centralized algorithm in which a leader is elected, who then allocates WBSNs to frequencies. However, such algorithms would require substantial additional protocol functions, including for leader election (which would also raise a lot of complex questions around selfishness, trust and fairness), signaling protocols allowing a newly arrived WBSN to be allocated a channel, and signaling

protocols for graceful release of channels or for periodic re-acquisition of a channel. It appears likely that these additions require changes to the existing WBSN technologies (IEEE 802.15.4, IEEE 802.15.6), which is generally undesirable and might lead to interoperability problems.

On the theoretical side this paper contributes novel and exact abstract Markovian models for discrete-time (or round-based) repeated balls-in-bin models, which may also have applications in other fields of networking and distributed systems.

1.2 Related Work

Broadly, this paper is in the area of frequency or channel allocation for wireless networks [7], [1]. In contrast to technologies like Bluetooth or the TSCH variant of IEEE 802.15.4 which apply frequency hopping all the time, the IEEE 802.15.6 standard and the original IEEE 802.15.4 standard (on which ZigBee is based) follow a model in which a WBSN will normally stay on the same channel throughout, unless higher layers make a decision to switch the channel. This is the model considered in this paper. The IEEE 802.15.6 standard [19] foresees a mechanism for channel-hopping, for the IEEE 802.15.4 standard such a mechanism has been described in [40]. Here, a WBSN observes its own performance and switches to another frequency channel in the 2.4 GHz band when the performance is unsatisfactory. The performance evaluation has been carried out for the case of external interference coming from WiFi interferers. A similar mechanism has been used in [26] for a scenario in which many co-located IEEE 802.15.4 networks have to share the same channel resources in the 2.4 GHz band and create so-called internal interference to each other. The results show that with autonomous channel adaptation a better utilization can be achieved and more WBSNs achieve satisfactory packet loss performance. It should be noted that frequency adaptation is not the only mechanism that has been considered to deal with internal performance. Other proposals suggest to adapt the parameters of the MAC protocol, for example the beacon and superframe order in IEEE 802.15.4 [33], [28], [32], or the parameters of the backoff process [36], [8]. Power adaptation has been considered in [39]. Other frequency allocation or adaptation algorithms are for example discussed [10] (a follow-up on [9]), which integrates frequency allocation with adjustment of transmission phases, and [31], [13], which propose an algorithm somewhat similar to the RJS algorithm, but with jumping probabilities that depend on the number of networks or nodes in the same channel. In our paper we only assume that an individual WBSN can tell whether it has sufficient channel quality or not, but it is not able to estimate the number of contenders, as that would require extra measurement procedures.

Going beyond the confines of frequency allocation for WBSNs, there are many publications in which the allocation of a number K of stations to a number N of “slots” is considered, for example to time slots in medium access control protocols. A classical example of such a setting can be found in the PRMA protocol [15], in R-ALOHA [25], or more recently in the context of IEEE 802.11p/WAVE [17], [30]. A key difference between the problem considered in this paper and the problem considered in these other publications is that in the other schemes a station can monitor

all bins and, if it jumps, to pick only one of the bins which are either free or colliding. For example, in the PRMA protocol time is sub-divided into superframes, which are then further sub-divided into time-slots. A station picks a time slot for transmission, and if it was the only transmitter in this slot, it will properly receive an acknowledgment. This can be observed by all other stations, which will then avoid this time slot until it becomes free again. Decisions on a time slot to pick are made at the end of a superframe when all time slots have been observed. In the schemes considered in this paper, however, a node can always only observe its own bin and has no information at all about the other bins.

Balls-in-Bins problems (also often referred to as random allocations) have been widely considered in probability theory, see for example the monographs [16], [18]. In particular, results are available for the distributions of the minimum and maximum occupancy of an urn [14]. They also have found applications in wireless channel allocation problems [7], [1], load balancing problems for cache servers or cloud computing in the Internet [37], [35], [3], in task allocation problems in crowdsourcing applications [23], or in the analysis of hashing schemes [11]. In many of these papers an assumption is made that a “ball” looking for a “bin” has a chance to inspect the contents of a number of randomly chosen bins before making any decision, and then it will pick the most favorable (e.g. the least loaded) bin of these. This differs from our setting in that we assume that a WBSN can only observe its current channel and not any others.

In this paper we particularly consider the case of repeated balls-in-bins experiments, in which a random subset of colliding balls jumps again, until a non-colliding allocation has been reached, making the system under consideration an example of a (discrete-time) interacting particle system [24]. However, our setting is different from the standard models considered in the interacting particle systems literature (e.g. exclusion processes, contact processes, voter models), and to the best of our knowledge this kind of processes and their convergence time towards non-colliding states have not been widely considered in the literature.

1.3 Paper Overview

This paper is structured as follows: in the next Section 2 we introduce an exact discrete-time Markov chain model for the evolution of balls-in-bins allocations under the RJS algorithm, establish its transition probability matrix and the average hitting times, and provide a range of numerical results. Following this, in Section 3 we discuss the RJS-OB scheme and present results that suggest that it can achieve much smaller average times to reach non-colliding states than the basic RJS scheme. The sticky scheme is modeled in Section 4. In Section 5 we consider the behaviour of the RJS-OB scheme in an overloaded situation, i.e. where we have $K > N$. Our conclusions are given in Section 6.

2 $K \leq N$: An Exact Model for RJS

We assume slotted time. At the beginning of a time slot each WBSN checks whether it collides with one or more other WBSNs in the same channel. If not, the WBSN stays on the current channel for the remainder of the time slot. Otherwise, the WBSN – independently of other WBSNs – jumps out of its current channel with probability $p \in (0, 1)$ and picks any of the remaining $N - 1$ channels with equal probability. The jumping will happen shortly before the end of the current time slot. We will refer to p as the *jumping probability*. For convenience, from here on we will leave the specifics of WBSNs behind and adopt a more combinatorial model and language. We will refer to the different channels as *bins* and to the WBSNs as *balls*. Therefore, our system is related to random allocations [16], [14], but the presence of collisions, the introduction of a jumping probability, and the repeated throwing of a subset of the balls (the colliding ones) gives it unique characteristics.

The state space in the exact model captures all possible allocations of K balls to N bins, it is given by

$$\mathcal{S}_{N,K} = \left\{ \mathbf{s} = (s_1, \dots, s_N) : s_i \in \mathbb{N}_0, \sum_{i=1}^N s_i = K \right\},$$

i.e. the set of all N -vectors (s_1, \dots, s_N) with non-negative integer entries summing up to K . In such a state the component s_i indicates the number of balls in bin i . From simple combinatorial considerations the size of the state space is

$$|\mathcal{S}_{N,K}| = \left\langle \begin{matrix} N \\ K \end{matrix} \right\rangle = \frac{(N+K-1)!}{(N-1)! \cdot K!} \quad (1)$$

We partition the state space into two different classes of states: into non-colliding states

$$\mathcal{N}_{N,K} = \{ \mathbf{s} \in \mathcal{S}_{N,K} : s_i \leq 1, i = 1, \dots, N \}$$

(where for all states $\mathbf{s} = (s_1, \dots, s_N)$ in this class we have $s_i \leq 1$ for all i) and colliding states $\mathcal{C}_{N,K} = \mathcal{S}_{N,K} \setminus \mathcal{N}_{N,K}$. For $K \leq N$ the set of non-colliding states is nonempty and has size

$$|\mathcal{N}_{N,K}| = \binom{N}{K} \quad (2)$$

For the calculation of the state transition probabilities it is convenient to introduce the notion of the *type* of a state $\mathbf{s} = (s_1, \dots, s_N)$. This type is given by listing the numbers s_1, \dots, s_N sorted in decreasing magnitude and dropping the zeros at the end (if any). The type of a state \mathbf{s} is denoted by $T(\mathbf{s})$. As an example, the type of state $\mathbf{s} = (0, 2, 3, 7, 1, 0, 0, 0)$ is $T(\mathbf{s}) = (7, 3, 2, 1)$. Furthermore, we denote by $\mathcal{T}(\mathbf{s}) = \{ \mathbf{t} \in \mathcal{S}_{N,K} : T(\mathbf{t}) = T(\mathbf{s}) \}$ the set of all states that have the same type as state \mathbf{s} .

In the following, we will derive the transition probability matrix $\mathbf{P}(p)$ for the Markov chain with state space $\mathcal{S}_{N,K}$, and the entries of this matrix will be functions of the jumping probability p . We will then set up a (uniquely solvable when $K \leq N$) linear system for the average number of steps needed to reach a non-colliding state

when starting from any of the colliding states (this is also known as a **hitting time**, see [29, Sec. 1.3]). The number of equations is given by $|\mathcal{S}_{N,K}| - |\mathcal{N}_{N,K}|$ and grows rather quickly in N and K (compare Equations (1) and (2)). We discuss in Appendix A how one can reduce the system size substantially. The solutions for the hitting time in general depend on the starting state as well as on the parameter p .

2.1 Transition Probabilities

We consider the Markov chain governing the movements of the balls in the bins and derive its probability transition matrix $\mathbf{P} = \mathbf{P}(N, K, p) = [[p(\mathbf{t}|\mathbf{s})]]_{\mathbf{s}, \mathbf{t} \in \mathcal{S}_{N,K}}$, with entry $p(\mathbf{t}|\mathbf{s})$ being the probability to go from state \mathbf{s} to state \mathbf{t} .

Our aim is to find the probability of transitioning from one state $\mathbf{s} = (s_1, \dots, s_N) \in \mathcal{S}_{N,K}$ to a given successor state $\mathbf{t} = (t_1, \dots, t_N) \in \mathcal{S}_{N,K}$. Since we assume that for any non-colliding state $\mathbf{s} \in \mathcal{N}_{N,K}$ there are no further movements, we clearly have for $\mathbf{s} \in \mathcal{N}_{N,K}$ that:

$$p(\mathbf{t}|\mathbf{s}) = \begin{cases} 1 & : \mathbf{s} = \mathbf{t} \\ 0 & : \text{otherwise} \end{cases}$$

and hence the non-colliding states are absorbing.

Now consider a colliding start state $\mathbf{s} = (s_1, \dots, s_N) \in \mathcal{C}_{N,K}$. In this state \mathbf{s} there is at least one bin i for which $s_i > 1$. Without loss of generality we assume for the following discussion that the colliding bins are the first ones, followed by the non-colliding bins, i.e. there exists some $k^* \in \{1, \dots, N\}$ such that $s_i > 1$ for $i \leq k^*$ and $s_i \leq 1$ for $i > k^*$. In general, there may be many different movements of balls that might lead from state \mathbf{s} to state \mathbf{t} . Consider as an example $N = 10$ and $K = 8$, where the possible movements to get from state $\mathbf{s} = (0, 3, 0, 1, 0, 0, 4, 0, 0, 0)$ to state $\mathbf{t} = (0, 3, 0, 1, 1, 1, 2, 0, 0, 0)$ include the following:

- $(2, 5)(2, 6)(2, 7)(7, 2)(7, 2)(7, 2)$, where a pair (i, j) indicates that a ball is moved from bin i to bin j
- $(7, 5)(7, 6)$
- $(7, 2)(7, 2)(2, 5)(2, 6)$
- $(7, 2)(7, 5)(2, 6)$.

Note that all these different movements in general occur with different probabilities.

To calculate the overall transition probability, we decompose state $\mathbf{s} = (s_1, \dots, s_{k^*}, s_{k^*+1}, \dots, s_N)$ into $k^* + 1$ different vectors, namely $\mathbf{s}_1 = (s_1, 0, 0, \dots, 0)$, $\mathbf{s}_2 = (0, s_2, 0, \dots, 0)$, \dots , $\mathbf{s}_{k^*} = (0, \dots, 0, s_{k^*}, 0, \dots, 0)$ (which we will call the pure states) and the residual $\mathbf{s}_r = (0, \dots, 0, s_{k^*+1}, \dots, s_N)$ in which each entry is ≤ 1 . Below we show how to calculate the transition probability from a pure state (say, \mathbf{s}_1) to any of its possible successor states, the set of which we denote as \mathcal{P}_i for pure state \mathbf{s}_i . With this notation, and using standard vector addition, we can express the transition probability from state \mathbf{s} to \mathbf{t} as

$$p(\mathbf{t}|\mathbf{s}) = \sum_{\substack{\mathbf{t}_1 \in \mathcal{P}_1, \dots, \mathbf{t}_{k^*} \in \mathcal{P}_{k^*} \\ \mathbf{t}_1 + \dots + \mathbf{t}_{k^*} + \mathbf{s}_r = \mathbf{t}}} p(\mathbf{t}_1|\mathbf{s}_1) \cdot \dots \cdot p(\mathbf{t}_{k^*}|\mathbf{s}_{k^*}) \quad (3)$$

where we have used that the jumping decisions made by different balls are independent. The transition probability from one of the pure states, say state $\mathbf{s}_1 = (s_1, 0, \dots, 0)$ to one of its successor states, say $\mathbf{t}_1 = (t_1, \dots, t_N)$ with $t_1 + \dots + t_N = s_1$, is given by:

$$p(\mathbf{t}_1|\mathbf{s}_1) = (1-p)^{t_1} \cdot \left(\frac{p}{N-1}\right)^{s_1-t_1} \cdot \binom{s_1}{t_1, \dots, t_N} \quad (4)$$

where

$$\binom{n}{k_1, \dots, k_N} = \frac{n!}{k_1! \cdot \dots \cdot k_N!}$$

is the multinomial coefficient. The first factor accounts for the t_1 balls that have opted to stay within their current bin (with probability $1-p$ each). The second term accounts for the $s_1 - t_1$ balls that have opted to jump (with probability p) to one of the $N-1$ other bins (with each other bin having the same probability). The final multinomial coefficient counts the number of ways in which s_1 distinguishable balls can be partitioned so that the first bin gets t_1 balls, the second bin gets t_2 balls and so on.

Please note that in the case $K \leq N$ (i.e. at most as many balls as bins) the set of non-colliding states is non-empty, and that these states are absorbing. Furthermore, for $p \in (0, 1)$ all the colliding states are transient states, since they jump with positive probability into one of the noncolliding states from which they never come back. This implies that the Markov chain will with probability one reach one of the non-colliding states when started in any state [29].

2.2 Hitting Time Calculation

We proceed towards the calculation of the key performance metric for the case $K \leq N$, which is the average number of steps (or the average time) to reach a non-colliding state $\mathbf{t} \in \mathcal{N}_{N,K}$ from some starting state $\mathbf{s} \in \mathcal{C}_{N,K}$. This time is also known as a *hitting time*, in our case for the set of non-colliding states. It is well known that the vector $(k_{\mathbf{s}} : \mathbf{s} \in \mathcal{S}_{N,K})$ of hitting times for all start states $\mathbf{s} \in \mathcal{S}_{N,K}$ solves the following system of linear equations ([29, Thm. 1.3.5]):

$$k_{\mathbf{s}} = 0 \quad : \mathbf{s} \in \mathcal{N}_{N,K} \quad (5)$$

$$k_{\mathbf{s}} = 1 + \sum_{\mathbf{t} \in \mathcal{C}_{N,K}} p(\mathbf{t}|\mathbf{s}) \cdot k_{\mathbf{t}} \quad : \mathbf{s} \in \mathcal{C}_{N,K} \quad (6)$$

Note that [29, Thm. 1.3.5] and [29, Thm. 4.2.3] imply that the solution to Equations (5) and (6) exists, is unique and non-negative. Unfortunately, even for small to moderate values of N and K the size of this linear equation system grows quickly, which makes it hard to solve in practice. For example, for $N = 15$ and $K = 10$ the full system comprises of $|\mathcal{S}_{15,10}| - |\mathcal{N}_{15,10}| = 1,961,256 - 3,003 = 1,958,253$ unknowns.

In Appendix A we discuss a method which allows to drastically reduce the size of the linear equation system, and we have used this method to obtain the numerical results presented next. More precisely, the system size is given by the number of possible types for K minus one (excluding the type $(1, 1, \dots, 1)$). The number of

possible types for K balls corresponds to the number of partitions of an integer K , for which an explicit expression is not known (see [14, Sect. I.3.1], see also OEIS A000041¹), but asymptotically (for large K) this number is approximately [14, Sect. I.3.1]:

$$\frac{1}{4K\sqrt{3}} \cdot \exp\left(\pi\sqrt{\frac{2K}{3}}\right) \quad (7)$$

In our example with $N = 15$ and $K = 10$ the system size reduces to 41 unknowns (see Appendix A). However, this method does not eliminate the calculation of all transition probabilities according to Equation 3, which remains a limiting factor.

2.3 Numerical Results

For the case of $K = 2$ it is straightforward to establish and solve the system (5) for a generic value of the jumping probability p . There is only one colliding type, the hitting times for all colliding states are identical, and it is possible to find the value of p which minimizes the common hitting time as

$$p^* = \frac{N-1}{N}$$

so that the common hitting time becomes

$$h(p^*) = \frac{N}{N-1}$$

Note that for larger values of N (i.e. more empty bins) the optimal jumping probability p^* increases – i.e. the more free bins there are, the more aggressively colliding nodes should jump. However, the case $K = 2$ is deceivingly simple and already for $K = 3$ new phenomena occur. First, the average hitting time is in general different for two colliding states of types (3) and (2, 1). In particular, for the considered values of N the best achievable hitting time for type (3) is generally larger than the best achievable hitting time for type (2, 1). Secondly, for both types (3) and (2, 1) the optimal jumping probability increases as N increases, with more free space it makes sense to jump more aggressively. But for each considered value of N , the optimal jumping probability for type (3) is smaller than for type (2, 1), i.e. balls from more crowded bins should jump less aggressively than balls from less crowded bins. Note that for $K \geq 3$ we have not been able to find closed-form expressions for the optimal jumping probability, as it involves finding roots of polynomials of fifth and higher degree [12, Chap14].

In Figure 1 we show results for the two cases $N = K = 5$ and $N = K = 7$, for the jumping probability p ranging from 0.04 to 0.99 in steps of 0.01, and a starting state where all balls are placed in the same bin. The figure shows the average hitting time versus p , obtained both numerically from the Markov model, and from running a

¹ OEIS is the online encyclopedia of integer sequences, see <https://oeis.org>.

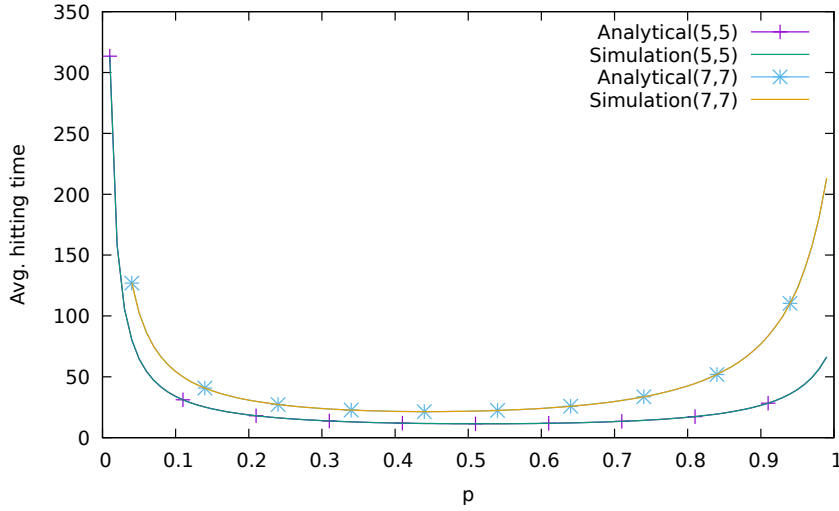


Fig. 1 Hitting times for varying p for a system with $N = K = 5$ and $N = K = 7$ (all balls start in the same bin).

custom-made Monte-Carlo simulator.² In the simulation, we have carried out 20,000 replications of the experiment for each value of p , and in the figure we report the average of these. There are two important conclusions: the first is that simulation and the analytical model show excellent agreement (the curves for the simulation and analytical results for the same value of $N = K$ overlap perfectly), which enhances trust in the validity of either of these. Secondly, it can be seen that the actual choice of p has a substantial impact, with average hitting times for $N = K = 5$ ranging from a few hundreds of time steps (for small p) down to a minimum of ≈ 11.4 time steps, similarly for $N = K = 7$.

In Figure 2(a) we show for varying N and $K \in \{N, N - 1\}$ and for different choices of the type of initial state the optimal jumping probabilities, i.e. the probabilities that achieve the smallest average hitting times. These probabilities have been obtained numerically from the Markov model by considering all p between 0.01 and 0.99 with a spacing of 0.01. In Figure 2(b) we show the resulting best hitting times. The following observations hold:

- When increasing N while keeping the difference $N - K$ fixed, the optimal p decreases with increasing value of N , while the resulting optimal hitting time increases with N (and K) at a superlinear rate.

² The simulator has been written in the Haskell programming language. Its operation is conceptually simple: it starts with an allocation of all K balls into the same urn and then proceeds in rounds. At the start of a round it checks whether there are any urns with more than one ball (collisions). If this is not the case the simulation stops and the simulator outputs the number of total number of rounds required. If there is at least one urn with a collision then for all collided balls we run an independent random experiment to determine whether the ball jumps, and if so, a new urn will be chosen for the ball with uniform probability (excluding the urn the colliding ball was in). Then the next round starts.

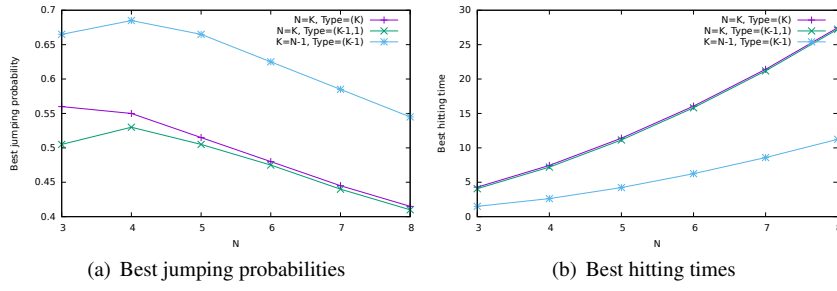


Fig. 2 Best jumping probabilities / hitting times for varying N , K , different start types

- Clearly the optimal p depends on this difference $N - K$, and p should be chosen larger as this difference increases.
- From Figure 2(b), the best hitting time for the type (K) is generally slightly larger than the best hitting time for type $(K - 1, 1)$. We hypothesize that generally the type (K) has the worst hitting time of all types and can thus be regarded as a “worst-case” type, but for increasing K this difference vanishes.
- We can also confirm the previously observed trend that for a given N the hitting times increase with K .

Unfortunately, besides identifying these trends, the complexity of the Markov model for RJS makes it very difficult to find a closed-form expression for the optimal value of p for given N and K .

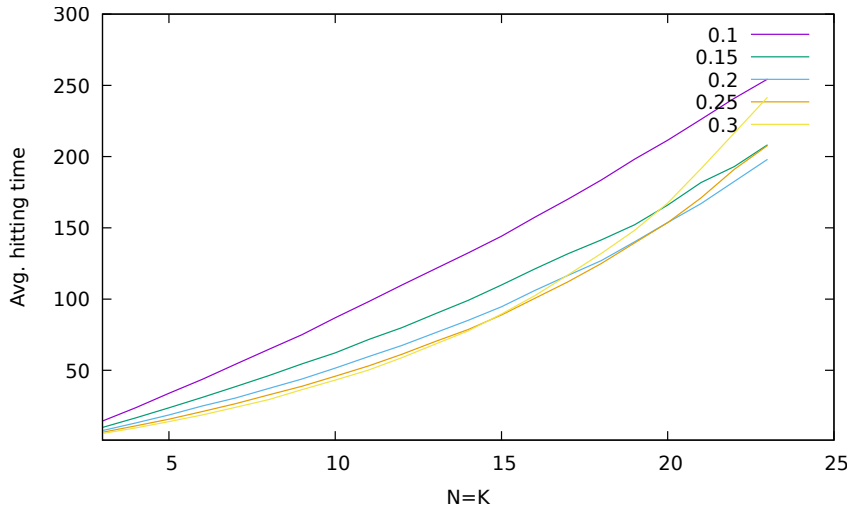


Fig. 3 Average hitting times for varying $N = K$ and start type (K)

Max. bin occupation	# Occurences	Percentage
2	14,568	84.06
3	2,666	15.38
4	65	0.38
5	3	0.02
6	3	0.02
≥ 7	26	0.15

Table 1 Summary statistics for one single simulation run

N	“Natural” Scheme	RJS ($p = 0.1$)
2	2.00	5.55
4	8.87	23.67
6	30.83	43.23
8	107.43	64.47
10	385.14	86.52
12	1485.39	110.53
14	6024.79	133.18

Table 2 Simulated average hitting time for the RJS scheme and the “natural” scheme for varying N

In Figure 3 we consider for different fixed values of p the hitting times for the worst-case type (N) for varying $N = K$. We have considered values of p from 0.1 to 0.9 with a spacing of 0.05, but we only show curves for p up to 0.3, as larger values of p lead to very large hitting times for increasing N . These averages have been obtained by simulation, each point is the average of 20,000 replications. The figure suggests that:

- The average hitting time grows quickly in N for fixed p .
- As N grows, the value of p that gives the smallest hitting time becomes smaller, e.g. for $N \leq 10$ the best value of p is 0.4, for $11 \leq N \leq 14$ the best value is 0.3 and so forth. We hypothesize that the optimal jumping probability for $N, K \rightarrow \infty$ while keeping $K \leq N$ will approach zero asymptotically.

In Table 1 we provide for $N = K = 175$, starting type (175) and $p = 0.1$ some summary statistics of one particular simulation run, where we record for each step in the simulation the number of balls in the most occupied bin and count how often a certain maximum bin occupation has occurred. It can be seen that $\approx 99.4\%$ out of a total of 17,331 steps taken have a maximum occupation of either 2 balls ($\approx 84\%$) or 3 balls ($\approx 15.4\%$). This suggests that for larger values of N the influence of the type of the starting state (here: the worst-case type (K)) is negligible and most of the time is spent in states with several bins having two or three balls only.

We finally consider a natural question about RJS: can we find something even simpler? A “natural” candidate scheme would be a scheme in which a colliding ball actually jumps with probability one (and not with some jumping probability p) and is allowed to jump into *any* bin with equal probability, including the bin it currently resides in. We have simulated such a scheme and compared it to RJS configured with a jumping probability of $p = 0.1$. The average hitting time results for varying

$N = K$ and taken over 1,000,000 replications are shown in Table 2. It is evident that the “natural” scheme shows very poor performance rather quickly.

3 $K \leq N$: A One-Bit Extension

The RJS algorithm discussed so far has desirable design properties (it does not require any coordination and converges with probability one to a non-colliding state when $K \leq N$) but the hitting time performance for larger values of N and K appears to have room for improvement. We present an algorithm which retains those desirable properties and achieves much better hitting times, at the expense of three additional configuration parameters, which are probability values, and one additional bit of state besides the (dynamically changing) jumping probability p in each node. We refer to this algorithm as RJS-OB (where OB stands for “one bit”).

The basic idea of the proposed RJS-OB algorithm is that a ball which has “owned” a bin for at least one time slot (by which we mean that during this slot the ball was the sole occupant of this bin) is much more reluctant to leave this bin upon the next collision than another ball which has jumped into this bin. The RJS-OB scheme works as follows:

- Each node i carries one additional bit, denoted as o_i (for “ownership”) and initialized to 0 or FALSE. The jumping probability p_i for this node is initialized to some value $q \in (0, 1)$.
- At the beginning of a time slot node i goes through two distinct steps:
 - In the first step it evolves its jumping probability p_i : if the current bin is non-colliding (i.e. node i is the sole occupant), it sets o_i to 1 (or TRUE) and its jumping probability to $p_i = q_O$, where $q_O \in [0, 1]$ is a parameter (the “owner probability”). If the bin is colliding and $o_i = 1$ (i.e. the node is the “owner”), it updates its jumping probability as $p_i := \min \{p_i + q_I, q_N\}$, where q_I and q_N are again fixed parameters (“probability increment” and “non-owner probability”). Otherwise, if the bin is colliding and $o_i = 0$, then p_i is set to $p_i = q_N$.
 - In the second step the node, if colliding, uses its newly updated jumping probability p_i to make a random decision on whether to jump to another bin. If so, each of the other $N - 1$ bins is chosen with equal probability and the flag o_i is set to 0 (or FALSE). The jump is executed shortly before the end of the time slot.

By setting the “ownership probability” q_O to a small value (e.g. $q_O = 0.05$), the increment probability q_I to a small value (e.g. $q_I = 0.01$) and the non-ownership probability q_N to a relatively large value (e.g. $q_N = 0.99$) we achieve a behaviour where a node that owns a slot jumps with only small probability (but growing after repeated collisions), whereas another node which just jumped into the same bin will jump again with high probability. We assess the impact of some of these parameters on the hitting time below.

We assess the performance of this scheme by simulation only, as the state space of a Markov model for RJS-OB would become intractably large, having to consider not only the allocations of nodes to bins, but also their state o_i and current jumping probability p_i .

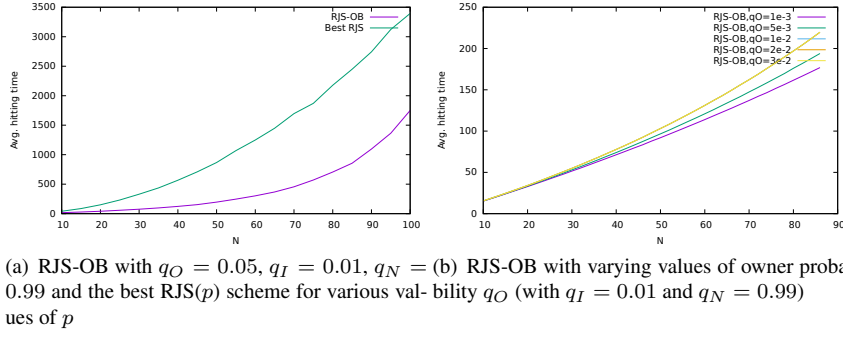


Fig. 4 Average hitting times for RJS-OB scheme

In Figure 4(a) we show the results for an experiment in which we have varied N and have assumed that $K = N$ holds. Furthermore, the system starts in the worst-case starting state of type (K) . We show the average hitting time (obtained over 5,000 replications, where in each replication we run rounds until a non-colliding state has been reached) for the RJS-OB scheme (with $q_O = 0.05$, $q_I = 0.01$, $q_N = 0.99$) and compare this against the best possible hitting time achievable with the RJS scheme after trying, for each N , various possible values of the jumping probability p (we have varied p from 0.01 to 0.4 in steps of 0.01). Clearly, the RJS-OB scheme is a significant improvement over the stateless RJS, which we attribute to the reluctance of balls “owning” a slot to jump away quickly after occasional collisions.

In Figure 4(b) we assess the impact of one particular parameter, the owner probability q_O while keeping $q_I = 0.01$ and $q_N = 0.99$ fixed. The hitting times have been obtained over 1,000,000 replications. It can be seen that smaller values of the owner probability (i.e. a stronger reluctance of a settled ball to leave its bin) lead to reductions of the hitting time and seem to approach a “linear” behaviour. We will return to this in Section 4.

4 $K \leq N$: The Sticky Variant

We have observed for the RJS-OB scheme described in Section 3 that for $K \leq N$ the average hitting time appears to approach a linear behaviour for smaller and smaller values of the ownership probability q_O as N and K increase (see Figure 4(b)). In this section we analyse a particular variant of the RJS-OB algorithm in which $q_O = 0$, $q_I = 0$ and $q_N = 1$, i.e. balls owning their slot never move away and non-owning colliding balls always jump away. We refer to this as the **sticky variant**, as owners stick to their bin. In this section we design and evaluate a Markov model of the sticky variant which, by making an inconsequential extra assumption, is vastly simplified as compared to the model set up for the RJS scheme and has a much smaller state space size.

To facilitate development of a simplified Markov model we make the assumption that a colliding non-owner ball can jump into **any** slot (with equal probability),

including the slot it currently resides in – this assumption removes the necessity to keep track of the particular bin in which a colliding non-owner ball resides.³ Note that for $K > N$ the sticky variant will be very unfair in the long-term, as N of the balls will eventually own a slot and the remaining $K - N$ balls will never settle. We will assume $K \leq N$ throughout this section.

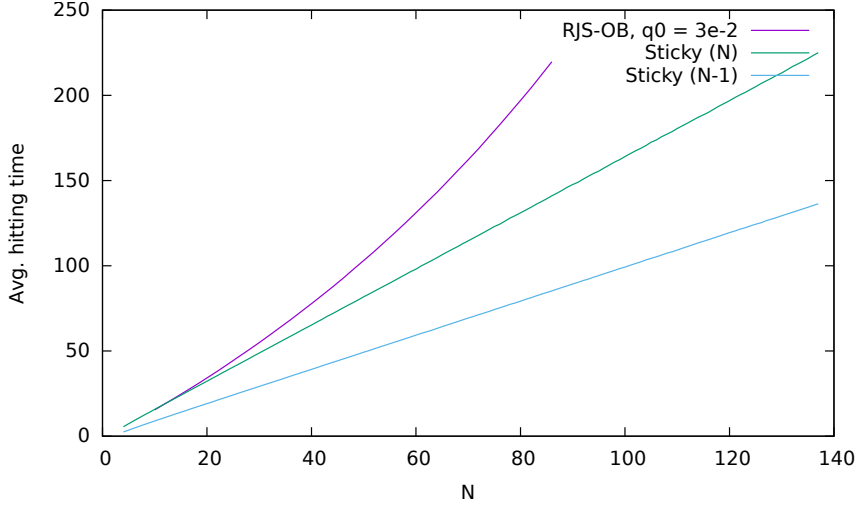


Fig. 5 Average hitting times for the RJS-OB scheme with owner probability $q_O = 3e - 2$ and the sticky scheme with $K = N$ balls and $K = N - 1$ balls

In Figure 5 we compare the average hitting times of the best RJS-OB scheme from Figure 4(b) against the sticky scheme for varying $N = K$ (all the times have been obtained by simulation). We have also added results for the sticky scheme with $K = N - 1$ balls. It can be seen that indeed the sticky scheme shows an almost linear scaling behaviour (it is not perfectly linear though), and furthermore that leaving away only one ball already provides substantial performance benefits. In the remainder of this section we develop a Markov chain model for the sticky variant, analyze some of its properties, and present a mixture of analytical and numeric results related to the asymptotic growth of the average hitting time for increasing N and K .

4.1 Markov Model for the Sticky Variant

Since a colliding ball can jump into any bin, there is no need to keep track of the bin in which a colliding ball currently resides. Therefore, it suffices to use the number of owning or “settled” balls as state variable – recall that a ball is settled in a bin

³ We have furthermore confirmed through simulations (not reported here for space reasons) that already for modest values of N this approximation shows very similar performance to the case where a colliding ball is not allowed to jump into its current bin again.

when it was the sole occupant for at least one time slot (after which it claims ownership). The state space is $\mathcal{S} = \{0, 1, \dots, K\}$, and we denote the $(K+1) \times (K+1)$ state transition matrix by \mathbf{P} . Since the number of settled balls never decreases, $\mathbf{P} = [[p_{i,j}]]_{i,j \in \{0,1,\dots,K\}}$ is an upper-triangular matrix with $p_{K,K} = 1$, i.e. the state in which all balls are settled is absorbing. All other states are transient and have a non-zero probability to jump into the absorbing state, which guarantees that the absorbing state will be reached with probability one. The starting state is normally the state 0, i.e. the state in which there are not yet any settled balls. Note that the diagonal entries $p_{S,S}$ (for $S \in \mathcal{S}$) are the eigenvalues of \mathbf{P} .

The transition probabilities are derived in Appendix B using the framework of exponential generating functions. The probability to go from state S to state $S+T$ for $T \geq 0$ is given by:

$$p_{S,S+T} = \frac{\binom{N-S}{T} \cdot (K-S)! \cdot \left(\sum_{\nu=0}^{K-(S+T)} (-1)^\nu \binom{N-(S+T)}{\nu} \frac{(N-T-\nu)^{K-(S+T)-\nu}}{(K-(S+T)-\nu)!} \right)}{N^{K-S}} \quad (8)$$

which for $T = 0$ simplifies to:

$$p_{S,S} = \frac{(K-S)! \cdot \left(\sum_{\nu=0}^{K-S} (-1)^\nu \binom{N-S}{\nu} \frac{(N-\nu)^{K-S-\nu}}{(K-S-\nu)!} \right)}{N^{K-S}} \quad (9)$$

We have validated the analytical model for the state transition matrix \mathbf{P} against empirical transition matrices obtained from Monte-Carlo simulations.⁴ In Figure 6 we compare the average hitting time for the simulation model (where the average is taken over 1,000,000 replications for given N and K) and the analytical model. In both cases we start with 0 settled balls. It can be seen that they are in excellent agreement, as the curves for the simulation and analytical models overlap. However, the explicit forms of the transition probabilities (Equations (8) and (9)) are not particularly convenient.

4.2 Properties of the Transition Matrix

We here collect a few properties of the state transition probability matrix $\mathbf{P} = [[p_{i,j}]]_{i,j \in \mathcal{S}}$ for the sticky variant. First, the diagonal entries $\lambda_S := p_{S,S}$ are positive and strictly monotonically increasing in S , and we have $p_{K-1,K-1} = \frac{K-1}{N}$ and $p_{K,K} = 1$. This is verified in Appendices C and D. Hence, the upper-triangular transition matrix \mathbf{P} has $K+1$ distinct eigenvalues and is diagonalizable. Secondly, for $K \geq 3$ all entries above the diagonal are strictly positive, except for entry $[[P]]_{0,K-1} = 0$. This is verified in Appendix D.

⁴ The calculations for the analytical model of the transition matrix \mathbf{P} have been carried out with Mathematica[®] and are exact. The empirical transition matrix for given values of N and K has been estimated by simulating, for each start state s_0 from 0 to $K-1$, a number of one million one-step transitions and counting how often each possible successor state is assumed. From these counts we can calculate the transition probabilities. We have repeated this for $N = K$ ranging from $N = K = 10$ to $N = K = 150$ and have found excellent agreement between the calculated and the estimated transition matrices.

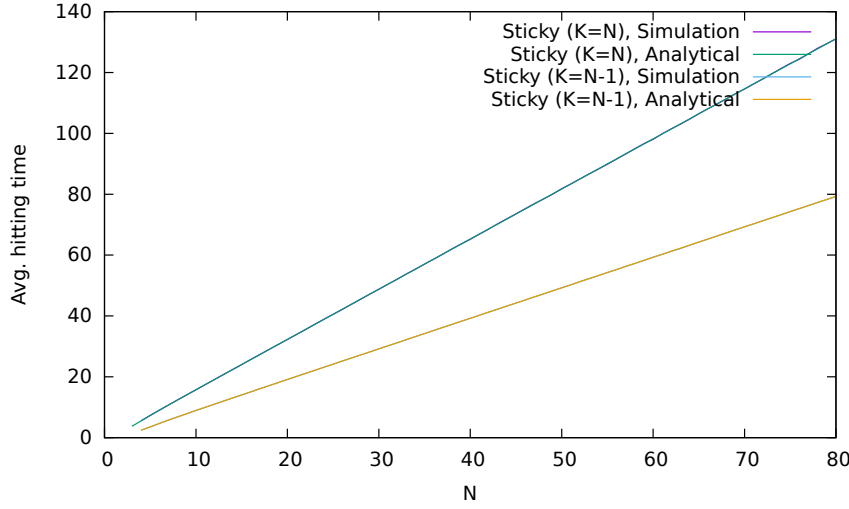


Fig. 6 Average hitting times for the Sticky variant, comparing simulation and analytical model (assuming $K = N$ balls).

Thirdly, from any start state $j \in \{0, \dots, K-1\}$ the speed of convergence to the single absorbing state K is at least geometric with rate $\lambda_{K-1}^n = \left(\frac{K-1}{N}\right)^n$, where $\lambda_{K-1} = p_{K-1, K-1} = \frac{K-1}{N}$ is the second-largest eigenvalue of \mathbf{P} . In other words, after n transitions the difference between one and the probability to be in the absorbing state K is bounded by λ_{K-1}^n times a constant independent of n . This is shown in Appendix E.

4.3 Asymptotics for Average Hitting Time

We conclude the investigation of the sticky variant by considering what happens for $N, K \rightarrow \infty$ (while $K \leq N$) and provide a simple bound as well as conjectures suggested by (limited) numerical evidence. In all of the following we assume that we start in state 0.

Given that from a state $S \in \mathcal{S}$ we can never move into a smaller state $S' < S$, a coarse upper bound $U_{N,K}$ on the average hitting time is given by the sum of all average state holding times:

$$U_{N,K} = \sum_{S=0}^{K-1} \frac{1}{1 - p_{S,S}} \quad (10)$$

where the average state holding time $\frac{1}{1 - p_{S,S}}$ results from the fact that the state-holding time in state S has a geometric distribution. In Figure 7 we compare this

upper bound against the exact analytical solution for the average hitting time.⁵ It can be seen that the upper bound is not tight, but shows an almost linear behaviour.

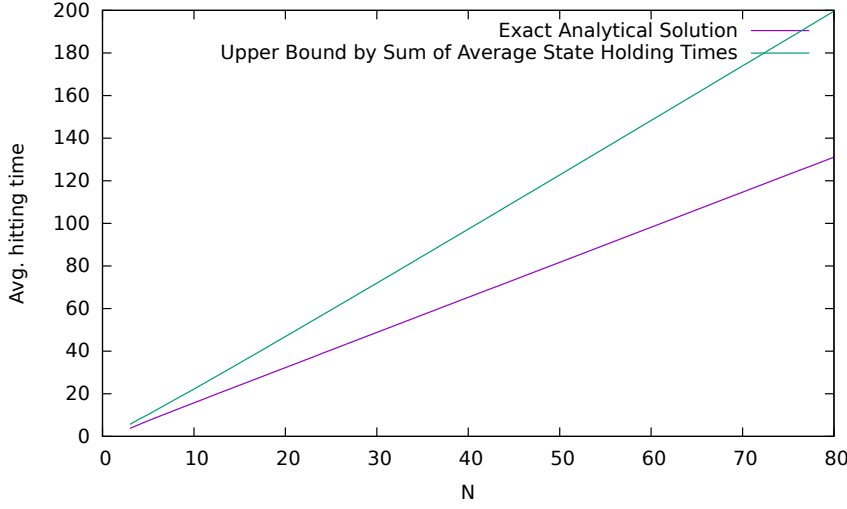


Fig. 7 Average hitting times for the Sticky variant, comparing analytical model and upper bound given by sum of state holding times (assuming $K = N$ balls).

We have conducted extensive numerical experiments (using exact calculations) to conjecture better upper bounds. Our starting point is Equation (29) for starting state $j = 0$. When we introduce the shorthand

$$C_{N,K,n} = - \left(\left(\frac{\lambda_0}{\lambda_{K-1}} \right)^n u_{0,0} v_{0,K} + \dots + \left(\frac{\lambda_{K-1}}{\lambda_{K-1}} \right)^n u_{0,K-1} v_{K-1,K} \right)$$

we get that

$$E[M_0] = \sum_{n=0}^{\infty} \lambda_{K-1}^n \cdot C_{N,K,n}$$

The numerical experiments suggest that:

- When for given N we consider all the possible values for $K \leq N$ then

$$C_{K,N,n} \leq \frac{N}{3}$$

resulting in

$$E[M_0] \leq \frac{N^2}{3(N-K+1)} \quad (11)$$

⁵ Both values have been obtained with Mathematica[®] using exact calculations. The analytical solution has been obtained from solving Equation (13) with \mathbf{P}' being the sub-matrix of \mathbf{P} in which the absorbing state K has been removed.

- For the case where we keep the difference between N and K fixed to $C = N - K$ we seem to have

$$C_{K,N,n} \leq C + 2$$

resulting in

$$E[M_0] \leq \frac{N(C+2)}{C+1} \quad (12)$$

These conjectures have been numerically confirmed for all $N \leq 50$. For larger values of N the computation time required by the exact calculations (which involve factors of the size of N !) quickly becomes prohibitive.

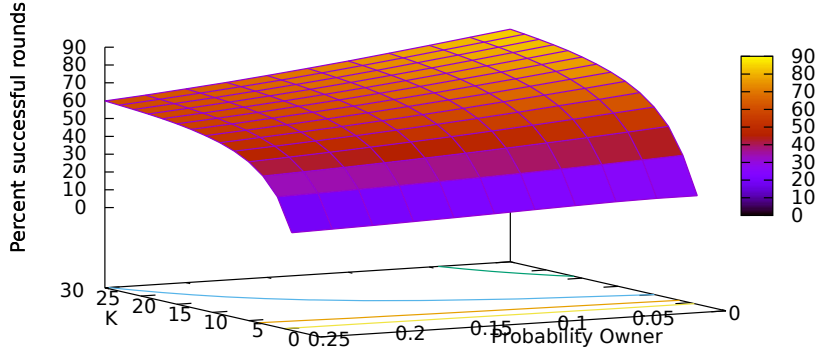
5 $K > N$: The One-Bit Extension in Overload Situations

The design of the RJS-OB scheme, and particularly the sticky variant, is geared towards reducing the average hitting time for the case $K \leq N$, i.e. with at most as many balls as there are bins. Clearly, this condition cannot be guaranteed in real applications and hence it becomes important to study the performance of the RJS-OB scheme in the scenario $K > N$ as well, where the hitting time is not meaningful anymore. For this study we use two main performance measures:

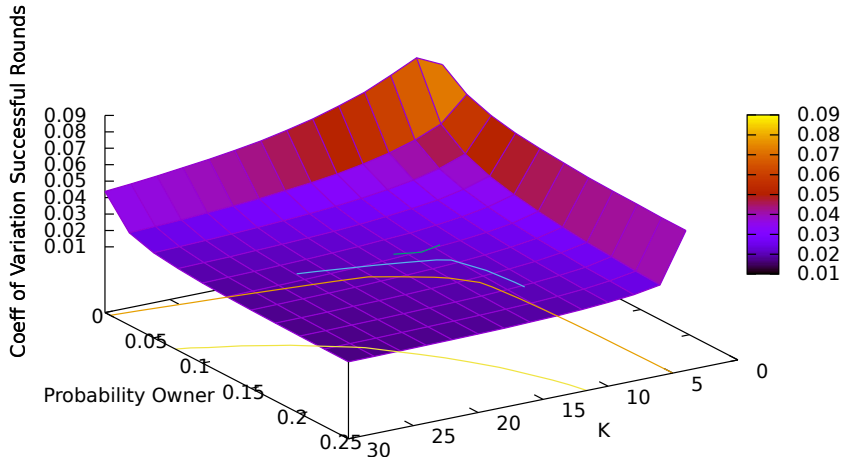
- The **average percentage of successful slots**, defined as follows: for a particular ball we measure the percentage of all rounds in which the ball is the sole occupant of a slot and can successfully use it for data transmission. The average is then taken over all the balls.
- By its construction and depending on its parameters, the RJS-OB scheme introduces differences between balls by making some “owners” of a bin and others not (particularly when the owner probability q_O and the probability increment q_I are configured to small values). It becomes possible that the owners receive a higher percentage of successful slots than the non-owners. To measure this unfairness we use the **coefficient of variation of the percentage of successful slots** (or simply coefficient of variation). In general, the coefficient of variation (CoV) of a positive random variable is defined as its standard deviation divided by the mean, and is a measure of variability. In our particular setting we measure the coefficient of variation of the percentage of successful slots of the different balls. Larger values of the CoV indicate larger differences within the node population.

We have instrumented our simulation model to measure these quantities. For all results we have used 10,000 rounds within a single simulation run, and for each set of parameters (we have varied N , K , p_O and p_N) we have carried out 10,000 independent replications, and in the following we report averages taken over these independent replications.

In the first set of results shown in Figures 8(a) and 8(b) we report both performance measures in a scenario where we have varied both N and K in a manner that N is always set to $K - 2$ and the owner probability q_O while keeping $q_I = 0.01$ and $q_N = 0.95$ fixed. It can be seen that the owner probability q_O has significant impact on the percentage of successful slots (Figure 8(a)), but also that the percentage of successful slots drops quickly as K becomes smaller (which means that the



(a) Average number of successful slots

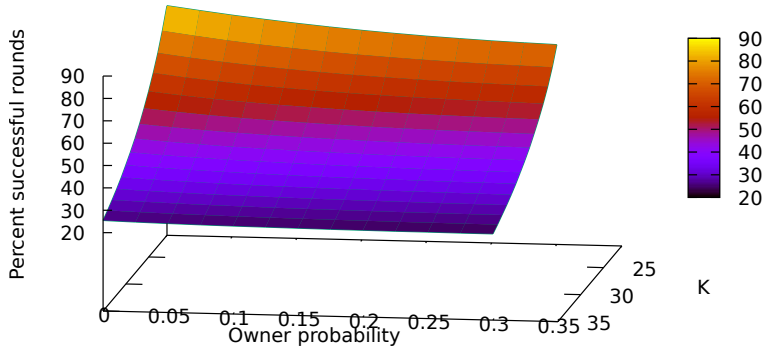


(b) Coefficient of variation

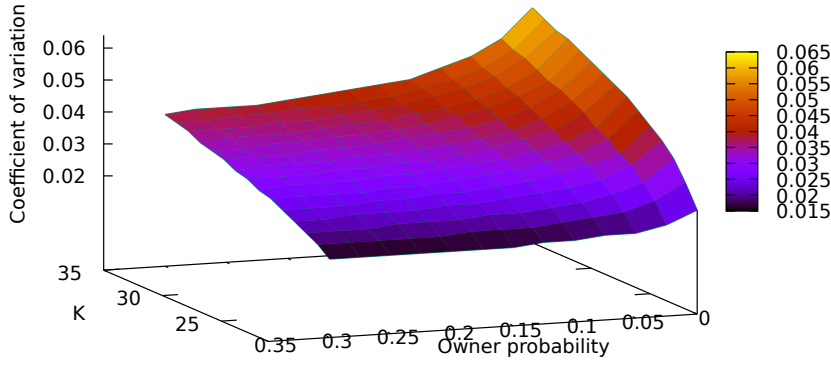
Fig. 8 Average number of successful slots and coefficient of variation for the RJS-OB scheme for varying K in an overload situation ($N = K - 2$) and varying q_O

ratio K/N increases). Furthermore, the results shown in Figure 8(b) confirm that the owner probability q_O has significant impact on the coefficient of variation, with smaller values for q_O being more “unfair” than larger values.

In the second set of results (shown in Figures 9(a) and 9(b)) we keep N fixed to $N = 20$ and vary K in the range $K \in \{21, \dots, 35\}$. We have varied the owner probability q_O in the range from 0.0 to 0.3 in steps of 0.025, and we have also var-



(a) Average number of successful slots



(b) Coefficient of variation

Fig. 9 Average number of successful slots and coefficient of variation for the RJS-OB scheme for $N = 20$ and varying K and q_O (each point is furthermore averaged over different values for q_N).

ied the non-owner probability q_N in the range from 0.1 to 0.5 in steps of 0.025. The probability increment q_I has been fixed to $q_I = 0.01$. However, for both the average number of successful slots and the coefficient of variation we have observed very little variation in our results when varying q_N , so the results shown in both figures are additionally averaged over all the q_N values. The results clearly show that as K (and with it the ratio K/N) increases, the percentage of successful slots decreases

substantially. Similarly, as the owner probability q_O increases, the percentage of successful slots decreases as well, although more moderately. Similarly, both parameters influence the coefficient of variation: increasing K increases the CoV, and decreasing the owner probability q_O increases the CoV as well, again highlighting the stronger unfairness when slot owners are too unwilling to give their slot up.

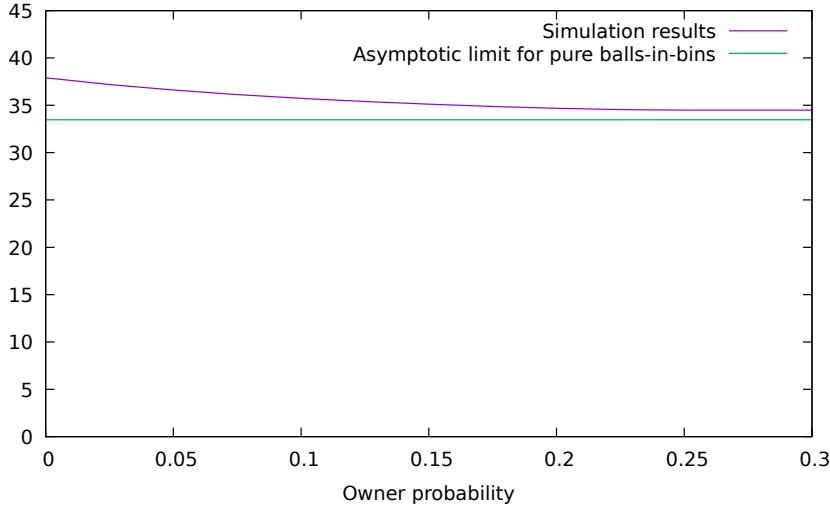


Fig. 10 Percentage of successful slots for $N = 20$, $K = 30$, $q_N = 0.25$ and varying q_O , comparing between RJS-OB and asymptotic value for static balls-in-bins allocation.

Finally, to show another perspective, in Figure 10 we show simulation results for the percentage of successful slots for the RJS-OB scheme with the non-owner probability fixed to $q_N = 0.25$, N fixed to $N = 20$, K fixed to $K = 30$, and varying owner probability q_O . We compare the RJS-OB scheme against the theoretical asymptotic average percentage of bins with exactly one ball when using a pure balls-in-bins allocation. This asymptotic estimate is known to equal $e^{-\frac{K}{N}} \cdot \frac{K}{N}$, see [14, p. 177]. These results suggest that by being unfair, the RJS-OB scheme can achieve somewhat better percentages of successful slots than a pure balls-in-bins allocation, but with increasing owner probability q_O (and thus increasing fairness) we approach from above the performance of pure balls-in-bins. This suggests an interesting item for future research, by providing balls (i.e. WBSNs) with the means to estimate the total number K of balls in the system (we suppose that N is usually known a-priori) and to its operational parameters q_N and q_O accordingly.

6 Conclusions

In this paper we have discussed the RJS, RJS-OB and the sticky schemes for the completely autonomous and distributed allocation of a number of WBSNs to a number

of frequency channels. We argue that the RJS-OB scheme is the most promising, as it has shown good performance for the considered parameters (and certainly superior performance over the RJS scheme) and can settle into a non-colliding state easily (provided there exists one), it is not restricted to the case $K \leq N$, and furthermore its behaviour can be adjusted for different population sizes.

There is significant potential for future work. For example, for the RJS-OB algorithm it would be interesting to develop good estimators for the number K of WBSNs present in the system (assuming that N is known a-priori) and to track this number when the population changes over time, e.g. due to WBSNs leaving or joining either as individuals or in batches. Next, assuming that K and N are known, it will be important to identify good values for the parameters q_O , q_N and q_I , so that in the case $K \leq N$ the average hitting time is minimized, whereas in the case $K > N$ we want to allocate undisturbed transmission opportunities in a fair manner. It will also be interesting to transfer the algorithm to a scenario without central slotted time, where each node makes decisions asynchronously (see for example [4]), or to situations where a frequency can host a variable number of WBSNs subject to a constraint that the total load does not exceed a threshold.

With respect to the sticky variant, it would be very interesting to confirm or refute the asymptotic results for the average hitting time of the sticky scheme (Equations (11) and (11)) which express the seemingly linear behaviour for fixed difference between N and K . Another interesting question is whether the RJS-OB or the sticky scheme can be improved upon by introducing more than one bit of additional state.

A Reducing System Size in Equation (5)

In this appendix we discuss how to reduce the size of the linear equation system (5) and (6), making it more accessible to numerical solution. The key roles are played by the notion of types introduced in Section 2 and the exploitation of symmetries.

To begin with, it is straightforward to see that we only need to consider terms for the colliding states in Equation (6), as the hitting times in non-colliding states are always zero. Next, consider two colliding states \mathbf{s}_a and \mathbf{s}_b of the same type. Therefore, there exists a permutation $\sigma(\cdot)$ of the bins from 1 to N such that $\mathbf{s}_b = \sigma(\mathbf{s}_a)$. From the definition of the transition probabilities (Equations (3) and (4)) it is straightforward to check that $p(\mathbf{t}|\mathbf{s}) = p(\tau(\mathbf{t})|\tau(\mathbf{s}))$ holds for any permutation $\tau(\cdot)$, and therefore the vector $(p(\mathbf{t}|\mathbf{s}_b) : \mathbf{t} \in \mathcal{C}_{N,K})$ of transition probabilities from state \mathbf{s}_b to all colliding states is a permutation of the vector $(p(\mathbf{t}|\mathbf{s}_a) : \mathbf{t} \in \mathcal{C}_{N,K})$, as applying the permutation $\sigma(\cdot)$ to all colliding states reproduces the entire set of colliding states.

Next, fix two different types $\mathcal{T}(\mathbf{s})$ and $\mathcal{T}(\mathbf{t})$. Since all states of the same type $\mathcal{T}(\mathbf{s})$ are permutations of each other, a similar argument shows that the vector of transition probabilities $(p(\mathbf{t}'|\mathbf{s}_a) : \mathbf{t}' \in \mathcal{T}(\mathbf{t}))$ from some state $\mathbf{s}_a \in \mathcal{T}(\mathbf{s})$ to all states in $\mathcal{T}(\mathbf{t})$ is a permutation of the transition probabilities $(p(\mathbf{t}'|\mathbf{s}_b) : \mathbf{t}' \in \mathcal{T}(\mathbf{t}))$ from any other state $\mathbf{s}_b \in \mathcal{T}(\mathbf{s})$ to all states in $\mathcal{T}(\mathbf{t})$. If we now order the states according to their type (i.e. all states of the same type are grouped together) and leave out the non-colliding states, then the matrix of state transition probabilities for the remaining states can be written as a block matrix of the following form:

$$\mathbf{P}' = \begin{bmatrix} \mathbf{A}_{(K)}^{(K)} & \mathbf{A}_{(K-1,1)}^{(K)} & \cdots & \mathbf{A}_{(2,1,\dots,1)}^{(K)} \\ \mathbf{A}_{(K)}^{(K-1,1)} & \mathbf{A}_{(K-1,1)}^{(K-1,1)} & \cdots & \mathbf{A}_{(2,1,\dots,1)}^{(K-1,1)} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{A}_{(K)}^{(2,1,\dots,1)} & \mathbf{A}_{(K-1,1)}^{(2,1,\dots,1)} & \cdots & \mathbf{A}_{(2,1,\dots,1)}^{(2,1,\dots,1)} \end{bmatrix}$$

where the matrix $\mathbf{A}_{\mathcal{T}_2}^{\mathcal{T}_1}$ contains all transition probabilities from states $\mathbf{s} \in \mathcal{T}_1$ to states $\mathbf{t} \in \mathcal{T}_2$, and in such a matrix all the rows are permutations of each other. Considering only the colliding states, the Equations

from (6) can be re-arranged as

$$(\mathbf{P}' - \mathbf{I}) \cdot \mathbf{k} = (-1, -1, \dots, -1)^T \quad (13)$$

where \mathbf{k} is the solution vector with the hitting times of all non-colliding states and \mathbf{I} is the identity matrix of suitable dimension. Please note that in this equation the matrix $(\mathbf{P}' - \mathbf{I})$ maintains the property of matrix \mathbf{P}' of being a block matrix made up of matrices where all the rows are permutations of each other: in all the “diagonal matrices” $\mathbf{A}_{\mathcal{T}}$ the respective diagonal elements are identical and remain so after subtracting the identity matrix. Furthermore, since we have omitted all transitions into non-colliding states (which occur with positive probability from every colliding state), the matrix \mathbf{P}' is strictly sub-stochastic, which in turn makes the matrix $(\mathbf{P}' - \mathbf{I})$ strictly diagonal-dominant and thus the equation system (13) is uniquely solvable. Note that its dimension is given by the size of the state space $|\mathcal{S}_{N,K}|$ with only the non-colliding states removed.

The key property which now allows to simplify the calculation of the hitting times is that all states of the same type have the same average hitting time. To see this, we argue as follows. If we denote by $\alpha_{\mathcal{T}_2}^{\mathcal{T}_1}$ the row sum of sub-matrix $\mathbf{A}_{\mathcal{T}_2}^{\mathcal{T}_1}$, then the system

$$\left(\begin{pmatrix} \alpha_{(K)}^{(K)} & \alpha_{(K-1,1)}^{(K)} & \dots & \alpha_{(2,1,\dots,1)}^{(K)} \\ \alpha_{(K-1,1)}^{(K-1,1)} & \alpha_{(K-1,1)}^{(K-1,1)} & \dots & \alpha_{(2,1,\dots,1)}^{(K-1,1)} \\ \alpha_{(K)}^{(2,1,\dots,1)} & \alpha_{(K-1,1)}^{(2,1,\dots,1)} & \dots & \alpha_{(2,1,\dots,1)}^{(2,1,\dots,1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{(K)}^{(2,1,\dots,1)} & \alpha_{(K-1,1)}^{(2,1,\dots,1)} & \dots & \alpha_{(2,1,\dots,1)}^{(2,1,\dots,1)} \end{pmatrix} - \mathbf{I} \right) \cdot \mathbf{k}' = (-1, -1, \dots, -1)^T \quad (14)$$

is again strictly diagonal-dominant and thus has a unique solution $\mathbf{k}' = (k_{(K)}, k_{(K-1,1)}, \dots, k_{(2,1,\dots,1)})$. Furthermore, its dimension is given by the number of types C_K existing for K balls (minus one, the non-colliding type $(1, 1, \dots, 1)$ – compare Equation (7)). We construct a vector \mathbf{k} from \mathbf{k}' by starting with repeating value $k_{(K)}$ exactly $|\mathcal{T}(K)|$ times, followed by $|\mathcal{T}(K-1, 1)|$ repetitions of $k_{(K-1,1)}$ and so forth. It is then clear that this vector \mathbf{k} is a solution of the full equation system (13).

Let us consider the savings that can be obtained from using system (14) instead of (13). For $N = 15$ and $K = 10$ we have that $C_{10} = 42$ and thus the system (14) has to be solved for 41 unknowns (after leaving out the single non-colliding type). As discussed before in Section 2.2, the full system (13) would have a dimension in the order of approximately 1.9 million unknowns.

B Sticky Variant: Derivation of Transition Probabilities

To derive explicit expressions for the transition probabilities of the sticky model we will use the framework of exponential generating functions (EGF) [14], [38], [27]. When applying EGF to balls-in-bin problems, a modified exponential power series

$$c(z) = \sum_{m=0}^{\infty} \frac{c_m}{m!} z^m \quad (z \in \mathbb{C}) \quad (15)$$

is associated to each bin, where the coefficients $c_m \in \{0, 1\}$ are chosen to express whether or not it is permissible to have m balls in the bin. The power series for several bins are then multiplied to get the overall EGF for the problem, and the m -th coefficient of the overall EGF gives the total number of allocations of balls to bins which satisfy all constraints simultaneously.

Exponential generating functions are one widely used class of generating functions, another class are ordinary generating functions. EGFs are appropriate in settings with labeled (i.e. distinct) entities. When given a general power series $f(z)$, we denote by $\{z^n\} f(z)$ the coefficient for the z^n term of $f(z)$, and for the particular case of an EGF $c(z)$ formed according to Equation (15), the coefficient c_m is given by $c_m = m! \cdot \{z^m\} c(z)$. Note that an EGF $c(z)$ (taken as a power series around $z = 0$) is an analytic / holomorphic function within its convergence radius, and it is a basic fact from complex analysis that we can recover the coefficients of a power series $c(z)$ from the derivatives of $c(z)$:

$$\frac{c_m}{m!} = \frac{(\partial_m c)(0)}{m!} \quad (16)$$

where ∂_m denotes the m -th complex derivative operator.

To transition from state $S \in \mathcal{S}$ to state $S + T \in \mathcal{S}$ we find the number of allocations of $K - S$ unsettled balls to bins such that

- Each of the S bins containing a settled ball can contain zero or more unsettled balls. To each such bin we assign the EGF e^z , in which all coefficients are one (compare Equation (15)).
- T of the bins that contained no settled ball before should now contain exactly one ball. To each of these bins we assign the EGF z (corresponding to $c_1 = 1$ and $c_m = 0$ for $m \neq 1$).
- The remaining $N - S - T$ bins each contain either zero or at least two balls. To each of these bins we assign the EGF $(e^z - z)$, excluding the case of exactly one ball.

The EGF corresponding to these constraints is given by

$$f(z) = (e^z)^S \cdot z^T \cdot (e^z - z)^{N-(S+T)} \quad (17)$$

and the expression $(K - S)! \cdot \{z^{K-S}\} f(z)$ gives the number of such allocations. This expression, however, refers to one particular choice of T bins out of the $N - S$ bins not containing a settled ball, and there are in total $\binom{N-S}{T}$ such choices. Therefore, the total number allocations of $K - S$ unsettled balls to bins satisfying the above constraints is given by

$$n(S, T) = \binom{N-S}{T} \cdot (K - S)! \cdot \left(\{z^{K-S}\} f(z) \right) \quad (18)$$

and since the total number of allocations of $K - S$ balls to N bins is given by N^{K-S} the transition probability becomes

$$p_{S, S+T} = \frac{n(S, T)}{N^{K-S}} = \frac{\binom{N-S}{T} \cdot (K - S)! \cdot \left(\{z^{K-S}\} f(z) \right)}{N^{K-S}} \quad (19)$$

To get explicit expressions, we expand the term $(e^z - z)^{N-(S+T)}$ of Equation (17) using the binomial theorem, and simplify the resulting expression (exploiting $K \leq N$) to arrive at:

$$n(S, T) = \binom{N-S}{T} \cdot (K - S)! \cdot \left(\sum_{\nu=0}^{K-(S+T)} (-1)^\nu \binom{N-(S+T)}{\nu} \frac{(N-T-\nu)^{K-(S+T)-\nu}}{(K-(S+T)-\nu)!} \right) \quad (20)$$

which for $T = 0$ simplifies to

$$n(S, 0) = (K - S)! \cdot \left(\sum_{\nu=0}^{K-S} (-1)^\nu \binom{N-S}{\nu} \frac{(N-\nu)^{K-S-\nu}}{(K-S-\nu)!} \right) \quad (21)$$

Note that a straightforward calculation yields that in particular we have

$$p_{K, K} = 1 \quad (22)$$

$$p_{K-1, K-1} = \frac{K-1}{N} \quad (23)$$

$$p_{K-2, K-2} = \frac{N + K^2 - 5K + 6}{N^2} \quad (24)$$

C Sticky Variant: Monotonicity of Self-Transition Probabilities

In this Appendix we show that the self-transition probabilities $p_{S, S}$ in the Markov model of the sticky variant (compare Equations (20) and in particular (9)) are strictly monotonically increasing in S for $K \geq 2$, i.e. that for $S \in \{0, 1, \dots, K-1\}$ we have $p_{S, S} < p_{S+1, S+1}$. Note that the self-transition probabilities are also all positive, which can be established probabilistically from the fact that colliding balls can

jump with positive probability back to exactly where they currently are, so that they remain in the colliding state.

The EGF corresponding to the diagonal entries can be specialized from Equation (17) for fixed state S to become:

$$f_S(z) = e^{zS} \cdot (e^z - z)^{N-S} \quad (25)$$

by setting $T = 0$, and valid for all $0 \leq S \leq K \leq N$. When taken as a power series, note that $f_S(z)$ has non-negative coefficients, since $g_1(z) = e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots$ and $g_2(z) = e^z - z = 1 + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots$ have only non-negative coefficients and $f_S(z)$ is formed out of $g_1(z)$ and $g_2(z)$ by multiplications. Note that the power series $g_3(z) = z$ also has only non-negative coefficients.

We can say more: for $S \neq 0$ a factor $g_1(z)$ is present which has strictly positive coefficients throughout, and all other factors of $f_S(\cdot)$ contribute a 1, so in $f_S(\cdot)$ there is at least one term of any order and all the coefficients are truly positive. For $S = 0$ all coefficients of $f_S(\cdot)$ of order zero or of order two or more are strictly positive. The coefficient of order one (for the term z) is zero, but this is relevant only for $K = 1$. So for all $K \geq 2$ and all S the coefficients of $f_S(\cdot)$ are strictly positive.

From the general expression for the transition probabilities (Equation (8)) and by taking into account that extracting coefficients is equivalent to taking derivatives (Equation (16)) we get

$$\begin{aligned} p_{S,S} &= \frac{(K-S)! \cdot \{z^{K-S}\} f_S(z)}{N^{K-S}} \\ &= \frac{(K-S)! \cdot \left[\frac{\partial_{K-S} f_S(0)}{(K-S)!} \right]}{N^{K-S}} \\ &= \frac{\partial_{K-S} f_S(0)}{N^{K-S}} \end{aligned} \quad (26)$$

With this, the condition $p_{S,S} < p_{S+1,S+1}$ is equivalent to the condition

$$\begin{aligned} &\partial_{K-S} \left[e^{zS} (e^z - z)^{N-S} \right] (0) \\ &< N \cdot \left[\partial_{K-S-1} \left[e^{z(S+1)} (e^z - z)^{N-S-1} \right] (0) \right] \end{aligned} \quad (27)$$

For $S = 0$ the left-hand side of this relation becomes

$$\partial_{K-1} \left[\partial_1 (e^z - z)^N \right] (0) = N \cdot \partial_{K-1} \left[(e^z - z)^{N-1} \cdot (e^z - 1) \right] (0)$$

and the right-hand side will be

$$N \cdot \left[\partial_{K-1} \left[e^z (e^z - z)^{N-1} \right] (0) \right]$$

Plugging this back into condition (27) after canceling N , subtracting the left-hand side from the right-hand-side and using the linearity of the derivative we get the condition

$$\begin{aligned} 0 &< \partial_{K-S} \left[e^z (e^z - z)^{N-1} - (e^z - z)^{N-1} \cdot (e^z - 1) \right] (0) \\ &= \partial_{K-S} \left[(e^z - z)^{N-1} (e^z - (e^z - 1)) \right] (0) \\ &= \partial_{K-S} \left[(e^z - z)^{N-1} \right] (0) \end{aligned}$$

which is just extracting the $K - S$ coefficient out of a power series (an integer power of $g_2(z)$) with coefficients that are non-negative in general and strictly positive from the z^2 coefficient onwards. Hence, for $N \geq K \geq 2$ and $S = 0$ the claim is true. For $0 < S \leq K - 1$ we get for the left-hand side of condition (27):

$$\begin{aligned} &\partial_{K-S-1} \left[\partial_1 \left[e^{zS} (e^z - z)^{N-S} \right] \right] (0) \\ &= \partial_{K-S-1} \left[S e^{zS} (e^z - z)^{N-S} + e^{zS} (N - S) (e^z - z)^{N-S-1} (e^z - 1) \right] (0) \end{aligned}$$

With this the condition to check becomes

$$\begin{aligned}
0 &< \partial_{K-S-1} \left[N e^{z(S+1)} (e^z - z)^{N-S-1} - S e^{zS} (e^z - z)^{N-S} - e^{zS} (N-S) (e^z - z)^{N-S-1} (e^z - 1) \right] (0) \\
&= \partial_{K-S-1} \left[e^{zS} (e^z - z)^{N-S-1} (N e^z - S (e^z - z) - (N-S) (e^z - 1)) \right] (0) \\
&= \partial_{K-S-1} \left[e^{zS} (e^z - z)^{N-S-1} (S z + (N-S)) \right] (0)
\end{aligned}$$

which again is built from (a sum of) products of $g_1(z)$, $g_2(z)$ and $g_3(z)$ and has strictly positive coefficients, verifying the claim.

D Sticky Variant: Strict Positivity of Transition Probabilities

In this Appendix we argue that the transition probabilities above the diagonal $p_{S,S+T}$ (with $T > 0$) in the Markov model of the sticky variant (compare Equations (20) and in particular (9)) are strictly positive, with one exception.

The starting point is again the EGF $f(z)$ from Equation (17), which is a product of integer multiples of the three elementary EGFs $g_1(z) = e^z$, $g_2(z) = e^z - z$ and $g_3(z) = z$.

The case of a state $0 < S < K$ is straightforward: the EGF $f(z)$ contains the factor z^T and the two other factors $(g_1(z))^S$ and $(g_2(z))^{N-(S+T)}$ both contribute a $z^0 = 1$ term, so the z^T term in $f(z)$ is strictly positive.

For the state $S = 0$ we distinguish three cases:

- With $T = K$ we are asking about state transition probability $p_{0,K}$ which from elementary combinatorial considerations is given by $N \cdot (N-1) \cdot \dots \cdot (N-K+1) > 0$.
- With $T = K-1$ we are asking that we are jumping out of a state in which no ball is settled into a state where every ball but one is settled – but to make this happen the first $N-1$ balls must jump alone into an empty bin (otherwise they would not settle) and the remaining ball cannot jump into the same bin as any of the others (otherwise none of the two would settle), so the last ball must jump into another empty bin, where it settles. More formally, for $S = 0$ and $T = K-1$ we get the generating function $f(z) = z^{K-1} \cdot (e^z - z)^{N-K+1}$ which for $K \geq 3$ does not contain a term in z , so the coefficient for z (and thus the transition probability into this state) are 0.
- For $1 \leq T \leq K-2$ the argument is very similar to the argument for $S > 0$.

E Sticky Variant: Convergence Speed towards Absorbing State

The claim of a geometric convergence speed of the Markov chain for the sticky variant towards the absorbing state 0 is in itself not a surprise, as it is similar to well-known results from the literature (compare for example [5, Chap. 6]), but the proofs of these frequently rely on the Perron-Frobenius theorem, which is only valid for irreducible transition matrices, which our upper-triangular matrix \mathbf{P} with the entries derived in Appendix B is not. We therefore proceed to verify this directly. Since \mathbf{P} is diagonalizable (compare Section 4.2), we can express \mathbf{P} as $\mathbf{P} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{U}^{-1} =: \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}$, where $\mathbf{D} = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_K)$ is a diagonal matrix made up of the eigenvalues / diagonal elements of \mathbf{P} and the columns of \mathbf{U} are given by the (linearly independent) right eigenvectors \mathbf{x}_i for eigenvalues λ_i .

Let M_j be a non-negative, integer-valued random variable denoting the number of steps it takes to get from starting state $j \in \{0, \dots, K-1\}$ to the absorbing state K . The average hitting time is then the expectation $E[M_j]$. For this expectation we can use the well-known so-called survivor representation [27]:

$$E[M_j] = \sum_{n=0}^{\infty} \Pr[M_j > n]$$

If we denote by $[v]_i$ the i -th component of a vector \mathbf{v} then we clearly have from the setup of our system that

$$\Pr[M_j > n] = [e_K - \mathbf{e}_j \cdot \mathbf{P}^n]_K = 1 - [e_j \cdot \mathbf{P}^n]_K$$

holds, where \mathbf{e}_k is the k -th unit vector in \mathbb{R}^{K+1} (taken as a row vector here). Recalling $\lambda_K = 1$ we get

$$\begin{aligned} & [e_j \cdot \mathbf{P}^n]_K \\ &= [e_j \cdot \mathbf{U} \cdot \mathbf{D}^n \cdot \mathbf{V}]_K \\ &= \lambda_0^n u_{j,0} v_{0,K} + \lambda_1^n u_{j,1} v_{1,K} \dots + \lambda_{K-1}^n u_{j,K-1} v_{K-1,K} + u_{j,K} v_{K,K} \end{aligned}$$

Since convergence to the absorbing state happens with probability one and recalling $\lambda_i < 1$ for $i \in \{0, 1, \dots, K-1\}$ we see

$$\begin{aligned} 0 &= \lim_{n \rightarrow \infty} [e_K - \mathbf{e}_j \cdot \mathbf{P}^n]_K \\ &= 1 - u_{j,K} v_{K,K} \end{aligned} \quad (28)$$

and therefore

$$\begin{aligned} & [e_K - \mathbf{e}_j \cdot \mathbf{P}^n]_K \\ &= -(\lambda_0^n u_{j,0} v_{0,K} + \lambda_1^n u_{j,1} v_{1,K} \dots + \lambda_{K-1}^n u_{j,K-1} v_{K-1,K}) \\ &= -\lambda_{K-1}^n \left(\left(\frac{\lambda_0}{\lambda_{K-1}} \right)^n u_{j,0} v_{0,K} + \dots + \left(\frac{\lambda_{K-1}}{\lambda_{K-1}} \right)^n u_{j,K-1} v_{K-1,K} \right) \\ &\leq \lambda_{K-1}^n \cdot \|\mathbf{u}_j\|_2 \cdot \|\mathbf{v}\|_2 \end{aligned} \quad (29)$$

(using the abbreviations $\mathbf{u}_j = (u_{j,0}, u_{j,1}, \dots, u_{j,K-1})$ and $\mathbf{v} = (v_{0,K}, v_{1,K}, \dots, v_{K-1,K})$). This verifies the claim.

References

1. Ahmed, E., Gani, A., Abolfazli, S., Yao, L.J., Khan, S.U.: Channel Assignment Algorithms in Cognitive Radio Networks: Taxonomy, Open Issues, and Challenges. *IEEE Communications Surveys and Tutorials* **18**(1) (2016)
2. Baker, S.B., Xiang, W., Atkinson, I.: Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access* **5**, 26,521 – 26,544 (2017)
3. Berenbrink, P., Brinkmann, A., Friedetzky, T., Meister, D., Nagel, L.: Distributing Storage in Cloud Environments. In: *Proc. IEEE 27th International Symposium on Parallel and Distributed Processing, Workshops and PhD Forum (IPDPSW)*. Cambridge, Massachusetts, USA (2013)
4. Berenbrink, P., Kling, P., Law, C., Mehrabian, A.: Tight Load Balancing via Randomized Local Search. In: *Proc. IEEE 27th International Symposium on Parallel and Distributed Processing (IPDPS)*. Orlando, Florida, USA (2017)
5. Bremaud, P.: *Markov Chains – Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, New York (1998)
6. Cavallari, R., Martelli, F., Rosini, R., Buratti, C., Verdone, R.: A Survey on Wireless Body Area Networks: Technologies and Design Challenges. *IEEE Communications Surveys and Tutorials* **16**(3) (2014). <http://www.comsoc.org/livepubs/surveys>
7. Chiochan, S., Hossain, E., Diamond, J.: Channel assignment schemes for infrastructure-based 802.11 WLANs: A survey. *IEEE Communications Surveys and Tutorials* **12**(1) (2010)
8. Choi, J.S., Zhou, M.: Performance analysis of zigbee-based body sensor networks. In: *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pp. 2427–2433. IEEE (2010)
9. Degesys, J., Rose, I., Patel, A., Nagpal, R.: DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks. In: *Proc. Symposium on Information Processing in Sensor Networks (IPSN '07)*. Cambridge, Massachusetts, USA (2007)
10. Deligiannis, N., Mota, J.F.C., Smart, G., Andreopoulos, Y.: Fast Desynchronization for Decentralized Multichannel Medium Access Control. *IEEE Trans. Communications* **63**(9), 3336–3349 (2015)
11. Demir, L., Kumar, A., Cuncu, M., Lauradoux, C.: The Pitfalls of Hashing for Privacy. *IEEE Communications Surveys and Tutorials* **20**(1) (2018). <http://www.comsoc.org/livepubs/surveys>
12. Dummit, D.S., Foote, R.M.: *Abstract Algebra*, 3rd edn. John Wiley, New York (2003)

13. Fischer, S., Maehoenen, P., Schoengens, M., Voecking, B.: Load Balancing for Dynamic Spectrum Assignment with Local Information for Secondary Users. In: Proc. 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN). IEEE, Chicago, Illinois, USA (2008)
14. Flajolet, P., Sedgewick, R.: Analytic Combinatorics. Cambridge University Press, Cambridge, UK (2009)
15. Goodman, D.J., Valenzuela, R.A., Gayliard, K.T., Ramamurthi, B.: Packet reservation multiple access for local wireless communications. IEEE Trans. Communications **37**(8), 885–890 (1989)
16. Johnson, N.L., Kotz, S.: Urn Models and their Applications. John Wiley & Sons, New York (1977)
17. Kenney, J.B.: Dedicated Short-Range Communications (DSRC) Standards in the United States. Proceedings of the IEEE **99**(7), 1162–1182 (2011)
18. Kolchin, V.F., Sevast'yanov, B.A., Chistyakov, V.P.: Random Allocations. Vh Winston (1978)
19. LAN/MAN Standards Committee of the IEEE Computer Society: IEEE Standard for Local and metropolitan area networks – Part 15.6: Wireless Body Area Networks (2012)
20. LAN/MAN Standards Committee of the IEEE Computer Society: IEEE Standard for Low-Rate Wireless Networks – IEEE Std 802.15.4-2015 (2015)
21. Latre, B., Braem, B., Moerman, I., Blondia, C., Demeester, P.: A survey on wireless body area networks. Wireless Networks **17**, 1–18 (2011)
22. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society, Providence, Rhode Island (2009)
23. Li, Q., Yang, P., Tang, S., Xiang, C., Li, F.: Many is Better than All: Efficient Selfish Load Balancing in Mobile Crowdsourcing Systems. In: Proc. Third International Conference on Advanced Cloud and Big Data. Yangzhou China (2015)
24. Liggett, T.M.: Interacting Particle Systems. Springer, Berlin (2008)
25. Liu, T.K., Silvester, J.A., Polydoros, A.: Performance evaluation of R-ALOHA in distributed packet radio networks with hard real-time communications. In: Proc. 45th IEEE Vehicular Technology Conference (VTC), pp. 554–558 (1995)
26. Moravejosharieh, A., Willig, A.: Mutual Interference in Large Populations of Co-Located IEEE 802.15.4 Body Sensor Networks – A Sensitivity Analysis. Elsevier Computer Communications **81**, 86 – 96 (2016)
27. Nelson, R.: Probability, Stochastic Processes, and Queueing Theory – The Mathematics of Computer Performance Modeling. Springer Verlag, New York (1995)
28. Neugebauer, M., Plönnigs, J., Kabitzsch, K.: A new beacon order adaptation algorithm for IEEE 802.15.4 networks. In: Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on, pp. 302–311. IEEE (2005)
29. Norris, J.R.: Markov Chains. Cambridge University Press, Cambridge, UK (1997)
30. Park, Y., Kim, H.: Collision Control of Periodic Safety Messages With Strict Messaging Frequency Requirements. IEEE Trans. Vehicular Technology **62**(2), 843 – 852 (2013)
31. Petrova, M., Olano, N., Maehoenen, P.: Balls and Bins Distributed Load Balancing Algorithm for Channel Allocation. In: Proc. 7th IEEE/IFIP International Conference on Wireless On-demand Network Systems and Services (WONS), pp. 25 –30. Kranjska Gora, Slovenia (2010)
32. Qiang, L., Jun, T.: Minimum-energy-cost algorithm based on superframe adaptation control. In: Communications (ICC), 2011 IEEE International Conference on, pp. 1–5 (2011)
33. Sari, R., et al.: Analysis of the effect of beacon order and superframe order value to the performance of multihop wireless networks on IEEE 802.15.4 protocol. In: Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on, pp. 89–94. IEEE (2012)
34. Seneviratne, S., Hu, Y., Nguyen, T., Lan, G., Khalifa, S., Thilakarathna, K., Hassan, M., Seneviratne, A.: A Survey of Wearable Devices and Challenges. IEEE Communications Surveys and Tutorials **19**(4) (2017). <http://www.comsoc.org/livepubs/surveys>
35. Siavoshani, M.J., Pourmiri, A., Shariatpanahi, S.P.: Storage, Communication, and Load Balancing Trade-off in Distributed Cache Networks. IEEE Trans. Parallel and Distributed Systems **29**(4), 943 – 957 (2018)
36. Tao, Z., Panwar, S., Gu, D., Zhang, J.: Performance analysis and a proposed improvement for the IEEE 802.15.4 contention access period. In: Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE, vol. 4, pp. 1811–1818. IEEE (2006)
37. Wei Zhang and Timothy Wood and Jinho Hwang: NetKV: Scalable, Self-Managing, Load Balancing as a Network Function. In: Proc. IEEE International Conference on Autonomic Computing (ICAC). Wuerzburg, Germany (2016)
38. Wilf, H.S.: generatingfunctionology. Academic Press (1994)

-
39. Yazdi, E.T., Willig, A., Pawlikowski, K.: Coupling Power and Frequency Adaptation for Interference Mitigation in IEEE 802.15.4-Based Mobile Body Sensor Networks. In: Proc. IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). Singapore (2014)
 40. Yazdi, E.T., Willig, A., Pawlikowski, K.: Frequency Adaptation for Interference Mitigation in IEEE 802.15.4-Based Mobile Body Sensor Networks. Elsevier Computer Communications **53**, 102–119 (2014)